Stat 602 Homework 3

Kiegan Rice and Nate Garton

Due ??

Problem 9

kiegan will do this problem

Problem 10

Shown below is the matrix of the proportion of predictions that agree between any two methods. The row and column labeled "actual" corresponds to the true data label.

```
wine <- read.csv(file = "data/winequality-white.csv", sep = ";", check.names = TRUE)
y <- -1*(wine$quality <= 7) + 1*(wine$quality > 7)
wine$y <- as.factor(y)
train_wine <- wine[,-(ncol(wine) - 1)]

fitControl <- trainControl(method = "repeatedcv",
                           number = 10, repeats = 5, search = "grid")
#
# set.seed(1308)
mods <- c("knn", "nnet", "rpart","rf",
          "xgbTree", "C5.0","glmnet", "stepLDA",
          "svmLinear", "svmPoly", "svmRadial")
# preds <- list()
# fitmod <- list()
# for(i in 1:length(mods))
# {
#   fitmod[[i]] <- caret::train(y ~ ., data = train_wine, trControl = fitControl, method = mods[i])
# }
# names(fitmod) <- mods
#
# saveRDS(fitmod, file = "hw3_10_models.rda")
fitmod <- readRDS(file = "hw3_10_models.rda")
preds <- extractPrediction(models = fitmod)
pred_y <- data.frame("obs" = train_wine$y, "pred" = train_wine$y,
                     "model" = "actual", "dataType" = "Training", "object" = "none")

preds <- rbind(preds, pred_y)

pred_mat <- matrix(nrow = length(mods) + 1, ncol = length(mods) + 1)
mods[12] <- "actual"
for(i in 1:nrow(pred_mat))
{
  for(j in 1:nrow(pred_mat))
  {
    pred_mat[i,j] <- mean(preds[preds$model == mods[i],]$pred == preds[preds$model == mods[j],]$pred)
  }
}
```

```r
colnames(pred_mat) <- mods
rownames(pred_mat) <- mods
knitr::kable(round(pred_mat[,-ncol(pred_mat)], digits = 3))
```

	knn	nnet	rpart	rf	xgbTree	C5.0	glmnet	stepLDA	svmLinear	svmPoly	svmRadial
knn	1.000	0.990	0.990	0.967	0.969	0.985	0.990	0.990	0.990	0.990	0.990
nnet	0.990	1.000	1.000	0.963	0.965	0.987	1.000	1.000	1.000	1.000	1.000
rpart	0.990	1.000	1.000	0.963	0.965	0.987	1.000	1.000	1.000	1.000	1.000
rf	0.967	0.963	0.963	1.000	0.998	0.976	0.963	0.963	0.963	0.963	0.963
xgbTree	0.969	0.965	0.965	0.998	1.000	0.978	0.965	0.965	0.965	0.965	0.965
C5.0	0.985	0.987	0.987	0.976	0.978	1.000	0.987	0.987	0.987	0.987	0.987
glmnet	0.990	1.000	1.000	0.963	0.965	0.987	1.000	1.000	1.000	1.000	1.000
stepLDA	0.990	1.000	1.000	0.963	0.965	0.987	1.000	1.000	1.000	1.000	1.000
svmLinear	0.990	1.000	1.000	0.963	0.965	0.987	1.000	1.000	1.000	1.000	1.000
svmPoly	0.990	1.000	1.000	0.963	0.965	0.987	1.000	1.000	1.000	1.000	1.000
svmRadial	0.990	1.000	1.000	0.963	0.965	0.987	1.000	1.000	1.000	1.000	1.000
actual	0.967	0.963	0.963	1.000	0.998	0.976	0.963	0.963	0.963	0.963	0.963

Problem 11

kiegan will do this problem

Problem 12

The predictors from the boosted tree, kNN, and C5.0 are all better than the worst predictors (which all had tied training error of 0.963), and seem to be the least correlated with one another. Therefore, we will include these three predictors in the generalized stacking model. The best cv error from the stacked model was ≈ 0.976 and the best individual model had a best cv error of ≈ 0.974. The two scores are very close, which is to be expected, as the correlation of the predictions between each of these models was still very high.

```r
set.seed(1308)
train_wine$target <- train_wine$y
levels(train_wine$target) <- c("bad", "good")
train_wine <- train_wine[,-(ncol(train_wine) - 1)]
# fitControl <- trainControl(method = "repeatedcv",
#                            number = 10, repeats = 5, search = "grid", classProbs = TRUE)
# clist <- caretList(target ~ .,data = train_wine,
#                    trControl = fitControl,
#                    methodList = c("xgbTree", "C5.0", "knn"))
# saveRDS(clist, file = "hw3_12_mod_list.rda")

clist <- readRDS(file = "hw3_12_mod_list.rda")

cverror <- c(clist$xgbTree$results[108,]$Accuracy,
             clist$C5.0$results[9,]$Accuracy,
             clist$knn$results[1,]$Accuracy)
stackmod <- caretStack(all.models = clist,
                       method = "rf",
                       metric = "Accuracy",
                       trControl = trainControl(
                         method = "cv",
```

```
                          number = 10,
                          search = "grid"
                        ))
```

```
## note: only 2 unique complexity parameters in default grid. Truncating the grid to 2 .
```

```
cverror <- c(cverror, stackmod$error[1,]$Accuracy)
names(cverror) <- c("xgbTree","C5.0","knn", "stack")
print(cverror)
```

```
##   xgbTree      C5.0       knn     stack
## 0.9742751 0.9659452 0.9628014 0.9759902
```